



AMYGDALA $z \mapsto z^2 + c$

A Newsletter of fractals & $\mathcal{M}$ (the Mandelbrot set)
AMYGDALA, Box 219, San Cristobal, NM 87564
505/758-7461

Current issues:

\$25.00 for ten issues (#13-22) (\$30 overseas)

\$45.00 for ten issues plus 24 color slides (\$50 overseas)

Back issues:

See Amygdala PRICE LIST & ORDER FORM

WORMTRACKS

—Ken Philip

A couple of weeks ago I received from Springer Verlag a copy (for review) of the new Mandelbrot/Julia program *The Game of Fractal Images*, by Marc Parment. [Review will follow in due course...] I found one delightful new feature in this program: a mode in which you could display superimposed on the Mandelbrot Set the iteration track of the point that the cursor was on. Since I had been working on iteration tracks myself at the time, plotting them directly on an otherwise blank screen, this feature appealed to me, and I began thinking about how to write my own version of a program to show iteration tracks *on* the Mandelbrot Set. Note: the iteration track is that actual series of locations in the complex plane that a point occupies as the equation $z \mapsto z^2 + c$ is iterated.

I wanted to be able to print the results to a LaserWriter, which meant the program would have to run in B&W. Furthermore, the interior of the Set would have to be white, rather than the normal black—but the region outside the Set should be white as well, since the iteration tracks have points both within and outside of the Set. What I needed was a way to plot only the *boundary* of the Set—and at that point I recalled the letter from G. A. G. Bennett in AMYGDALA #11 about his 'Mandelworm' algorithm. That routine would do just what I needed, provided I could implement it in True BASIC 2.0.

As he described 'Mandelworm', it was clear that it would fill in the corners of the boundary (see Fig. 1a). I preferred a boundary that could run diagonally (Fig. 1b), so I had to modify the algorithm so it could handle 8 primary directions rather than 4.

His algorithm started by plotting in the Set, and then rotated CCW until it passed outside, at which point it moved to the new pixel on the boundary and recycled. I decided to re-

Issue #14

November 4, 1988

Copyright © 1986,1987, 1988 Rollo Silver

CONTENTS

WORMTRACKS	1
TUTORIAL: MORE ON GEOMETRIC REPRESENTATION	4
THE SLIDES	5
PORTABLE SLIDE VIEWER	5
DIMENSIONS — Part 2	5
PHOTOGRAPHING THE COMPUTER SCREEN	7
AUTHORS	8
QUESTIONS	8
PRODUCTS	8
CIRCULATION	8

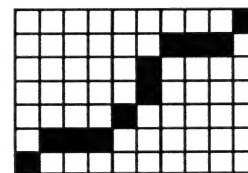
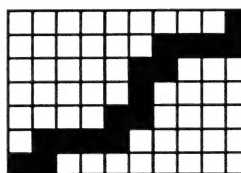


Fig. 1a

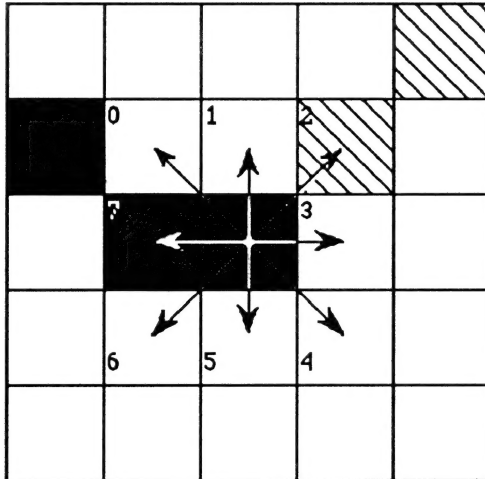
Fig. 1b

Small section of boundary

verse this procedure, and start plotting *outside* the Set, and swing CW until I hit a pixel on the boundary. In order not to miss any pixels, I had to devise a method to start hunting one step CW from the direction to the previously plotted boundary pixel. This was done using the MOD function, since by making all operations on the direction array MOD 8, then the variable 'dir+4' would refer to a direction 180° from 'dir', and 'dir+5' would be one step CW from that direction. So, after plotting a new boundary pixel, the operator 'dir+5' applied to the direction of that pixel would indicate the direction to start hunting from after the center of plotting had been moved to the new pixel. Figure 2 [next page] shows a 5x5 pixel area on the boundary (where black pixels have been plotted, striped pixels are to be plotted), and illustrates how this process works. The direction from the previous pixel to the current pixel is 3. MOD (3+5,8) is 0, which is the starting direction for the search for the next pixel (lying in direction 2).

A subroutine 'Worm' in the code implements this search

Figure 2 — Directions for pixel search



method, starting at $Re=-2$, $Im=0$, and plotting until Im goes negative at the east end of the Set. Since the + and - halves of the Set are symmetrical, both halves are plotted simultane-

ously. The dwell and escape radius may be set to whatever the user feels appropriate, within the constraint that the 'Worm' routine breaks down at a maximum dwell just over 500, and skips over the 'neck' between the head and the body of the Set, driving Im negative and stopping plotting. The grid is set to a spacing of .00625. The x/y coordinates of the starting point are carefully set to have this plot in precise register with the later iteration track plotting.

Now that we have the Set boundary, the next job is to plot iteration tracks. This is done within another subroutine, using various routines to iterate the basic equation, display the track, and read out coordinates to the screen. A GET MOUSE statement supplies the x/y coordinates of the cursor, which (when the button is held down) are transformed to Re/Im coordinates and sent to two nested subroutines which compute points on the iteration track provided they are not near overflow, and then display those points on the screen. The previous condition of the screen is saved in BOX KEEP, and when the button is released a BOX SHOW statement erases the current track by replotting that previous screen.

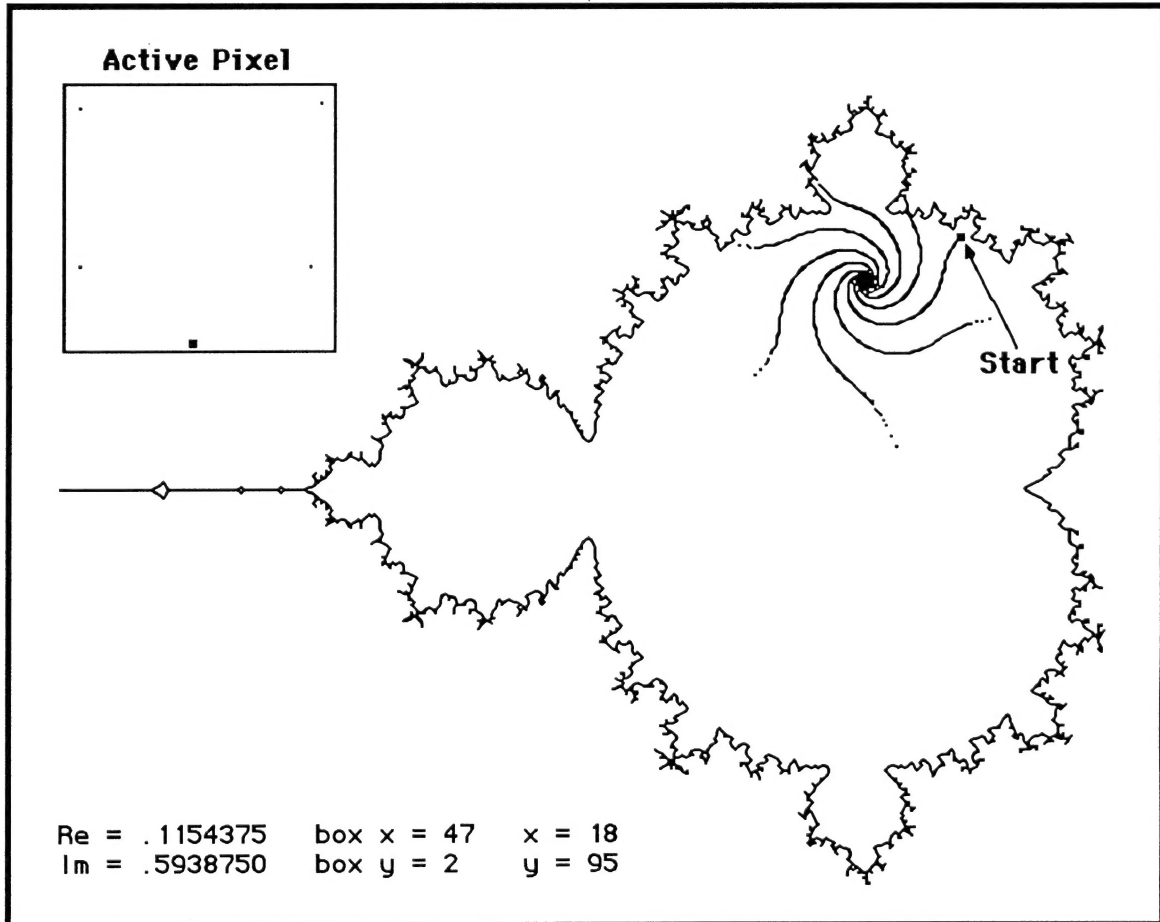


Figure 3 — Iteration track for point under NNE 'bud'

Note that True BASIC tries to be independent of screen pixels—but for this program we *want* to work directly with pixels. A SET WINDOW statement at the beginning of the program is therefore used to force True BASIC to work with the pixels on a 640x480 pixel screen (standard Mac II screen).

The .00625-step grid required for the image of the entire Set is too coarse to let you explore the finer details of the way iteration tracks can vary with position near the boundary of the Set—so a 100X magnify capability was added to the program. If you hold the button down and drag the cursor into the box to the upper left of the screen, the program switches to magnify mode and can compute and display 10,000 different iteration tracks for each original screen pixel, working on a .0000625 grid. Clicking the mouse to the right of or below that box (outside of a 10-pixel ‘guard zone’) will switch back to the normal 1X mode. While in magnify mode, the coordinate reporting adds x/y coordinates in the box to let you easily return to the exact same track at some later time without having to record 16-place *Re/Im* coordinates.

The box contents are also saved before entering magnify

mode and replotted on exit, which allows the program to leave markers in the box showing the locations that you have checked in magnify mode, and then restore the pristine box on exit. Both in the box and on the 1X screen, the starting point of an iteration track is marked with a 3x3-pixel square.

True BASIC will run on many different computers, so the code should be portable except for a few ‘MacTools*’ library routines which affect only the size of the type and the centering of the title over the box. Some machines will have trouble with the BOX KEEP/BOX SAVE statements, since BOX KEEP saves a graphic image as a string and may exceed the maximum string size on an MS/DOS machine. That problem may be solved by breaking the screen into two halves and saving and replotted each half independently. People wishing to write this program in some other language (like MS BASIC) will have to find another way to save and refresh the window.

The other possible problem is with the x/y coordinate reporting. True BASIC on the IBM does not seem to run in a 640x480-pixel mode, and therefore the y-coordinates will

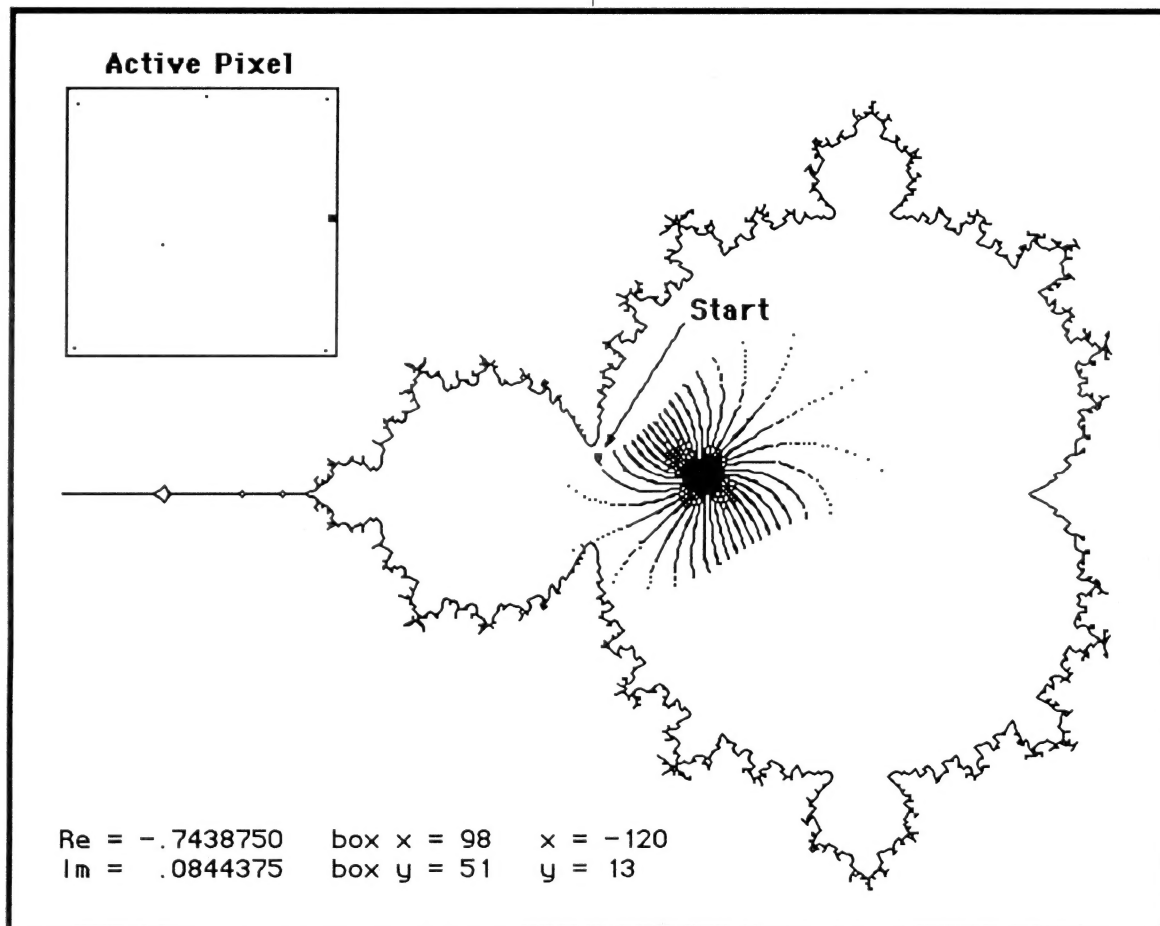


Figure 4 — Iteration track in Seahorse Valley

not be reported as integer numbers. There is no easy way around that except rewriting the program to cope with a smaller number of vertical pixels.

Figure 3 [page 2] and figure 4 [page 3] show sample iteration tracks produced with **Wormtracks**. The frame and the identification of the 'Start' point were added in MacDraw II.

Those who would like to see the source code for **Wormtracks** can obtain it from:

Kenelm W. Philip
1590 North Becker Ridge Road
Fairbanks, AK 99709

by sending a SASE and \$1 for a printed listing, or an SASE, \$1, and an initialized 3.5" disk for the file. Note that as supplied the program runs on a Macintosh II in True BASIC 2.01. On a smaller screen the program will still work, but the x/y coordinate reporting will not display integer values, and the text displaying routines will have to be changed for proper screen positioning and to avoid overwriting.

TUTORIAL: MORE ON GEOMETRIC REPRESENTATION

These tutorials form a more or less coherent series, beginning in Amygdala #1:

- Complex Numbers: The Basics (1.1, i.e. Amy #1, page 1)
- Tutorial: Basic Mathematics of M (2.2)
- Tutorial: Functions; Polynomials (4.2)
- Tutorial: Mandelbrot Polynomials (5.3)
- Tutorial: Conjugation (8.3)
- Tutorial: Inequalities (9.4)
- Tutorial: The Geometric Representation of Complex Numbers (10.2)

Let's continue with the geometric representation of complex numbers.

14.1 Geometric Multiplication

To get a geometric interpretation of the product of two complex numbers we consider polar coordinates, as first discussed in (1.1). The complex number $z = x + iy$ corresponds to the point (x, y) whose rectangular coordinates are x and y . The polar coordinates of the point are (r, θ) , where $x = r \cos \theta$, $y = r \sin \theta$; so we have

$$z = x + iy = r(\cos \theta + i \sin \theta) \quad (9)$$

This is the *trigonometric form* of the complex number z ; r is the modulus $|z|$, and is always ≥ 0 . The polar angle θ , expressed as $\arg \theta$, is called the *argument* or *amplitude* of z .

Now let's consider two complex numbers: $z_1 = r_1(\cos \theta_1 + i \sin \theta_1)$ and $z_2 = r_2(\cos \theta_2 + i \sin \theta_2)$. Applying the rule for multiplying complex numbers, we get:

$$\begin{aligned} z_1 z_2 &= r_1(\cos \theta_1 + i \sin \theta_1) r_2(\cos \theta_2 + i \sin \theta_2) = \\ &= r_1 r_2 (\cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 + \\ &\quad i (\cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2)) \end{aligned} \quad (10)$$

Applying the following well-known trigonometric identities to (10),

$$\cos(\theta_1 + \theta_2) = \cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2$$

$$\sin(\theta_1 + \theta_2) = \sin \theta_1 \cos \theta_2 + \cos \theta_1 \sin \theta_2$$

we get:

$$z_1 z_2 = r_1 r_2 (\cos(\theta_1 + \theta_2) + i (\sin \theta_1 + \theta_2)) \quad (11)$$

which product, per (9), has argument $(\theta_1 + \theta_2)$ and modulus $r_1 r_2$. Thus $|z_1 z_2| = |z_1| |z_2|$, and:

$$\arg(z_1 z_2) = \arg z_1 + \arg z_2 \quad (12)$$

This formula can be extended to arbitrary products, e.g. $\arg(z_1 z_2 z_3) = \arg z_1 + \arg(z_2 z_3) = \arg z_1 + \arg z_2 + \arg z_3$.

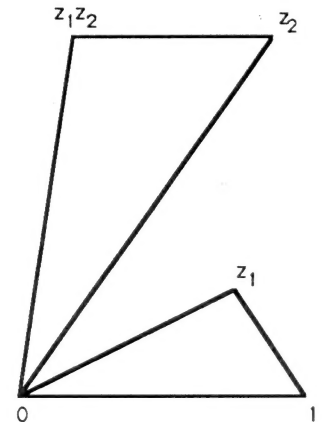
Therefore:

The argument of a product is equal to the sum of the arguments of its factors.

Now note first that $\arg 0$ is undefined, so (12) has meaning only if neither angle is 0. Second, the polar angle is defined only up to multiples of 360° : e.g. $\arg(1+i) = 45^\circ$, or -315° , or 405° , etc. If we use (12) to provide a geometric interpretation, we will not count multiples of 360° .

(12) can be used to develop a simple geometric construction for the product $z_1 z_2$.

The smaller triangle, with vertices $0, 1, z_1$, is similar to the larger triangle, whose vertices are $0, z_2, z_1 z_2$. The points $0, 1, z_1$, and z_2 being



given, this similarity determines the point $z_1 z_2$.

In the case of division, (12) is replaced by:

$$\arg(z_2/z_1) = \arg z_2 - \arg z_1 \quad (13)$$

For division the geometric construction is analogous, with the similar triangles being $0, 1, z_1$ and $0, z_2/z_1, z_2$.

EXERCISES FOR THE READER

E14.1. Prove that points z_1, z_2, z_3 are vertices of an equilateral triangle if and only if:

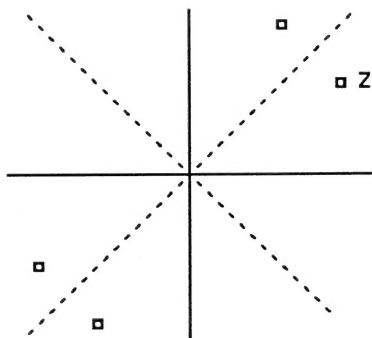
$$z_1^2 + z_2^2 + z_3^2 = z_1 z_2 + z_2 z_3 + z_3 z_1.$$

E14.2. Let z and w be two vertices of a square. Find the two other vertices (considering all possible cases).

E14.3. Find the center and the radius of the circle which circumscribes the triangle with vertices z_1, z_2, z_3 .

Express the result in symmetric form.

E14.4. Find the points symmetric to the point z with respect to the lines which bisect the angles between the coordinate axes: $\rightarrow$



THE SLIDES

Those subscribing to the color slide supplement now have C13: the fourth quad of slides of the current series of 24 slides.

181. John Dewey Jones: Red Dawn Over Key West. False-color satellite photo shows waves of communist influence spreading across the Caribbean to subvert Florida. M-view - 0.1013816-0.9567729i... 0.1012797-0.95639i.

185. John Dewey Jones: The Devil's Clock. Interesting branched structure; M-view: -0.1013432-0.9564415i...-0.1013199-0.9564255i.

481. Ken Philip: A nine-armed medusa or rosette, pale-blue against violet.

802. A.G. Davis Philip: Detail of the spike, in salmon and black.

PORTABLE SLIDE VIEWER

I've found a portable 35mm slide viewer that works quite well. The viewing lens is almost 3" square. The unit operates on 2 C batteries, holds up to 32 slides with a manual sliding feeder that automatically turns the power off when IN, and magnifies about 2.5 times. Weighs 1.5 lbs. It is available for \$14.98 from:

Lillian Vernon
510 South Fulton Avenue
Mount Vernon, NY 10550
914/633-6300

DIMENSIONS — Part 2

— Daniel Beisenherz

Hausdorff-Besicovitch Dimension

The Hausdorff-Besicovitch Dimension is the most interesting and helpful model for our purposes.

We now want to make an assertion about the length, volume, or area that various geometric objects cover. The length of Koch's Arc (one sixth of the Snowflake) is of infinite

length, but the line it is drawn by is to be imagined to be of no thickness. Does this mean that it fills no space at all ($\infty \times 0 = 0$???), or does it fill an area of infinite size ($\infty \times 0 = \infty$???)?

Although the new dimension we are about to discuss doesn't say that this fractal is such-and-such many centimeters long, it can give you a glimpse of how sophisticated this object is. The basic idea is to measure its length by means of some gauge (we'll see what this gauge possibly might look like). To measure a smooth curve, we'd need a ruler of one length-unit (you may think of one inch, one yard, one meter, or whatever). To measure an area a unit square would be of use, and to measure a body's volume we could make use of a unit cube (a cube whose edges are one unit long).

This would be sufficient to measure objects which are multiples of the gauge we possess — we didn't mark any partitions on it! For example, if we had a cube whose edges are three units long, we could put our unit cube into it nine times; but this does not mean that its Hausdorff-Besicovitch dimension would be nine! We are not finished yet!

To measure smaller, more complex objects, we have to derive smaller units from the gauge that we already possess. For that purpose, we divide the edges of the gauge into segments of equal length, say b segments. Thus the new gauge has length $r = 1/b$. To cover the unit ruler we need b new gauges; for the unit square we need b^2 ; for the unit cube we need b^3 new units of edge length r . Let's denote by N the number of new units we would need. Generally we can say:

$$N = b^D = (1/r)^D = r^{-D} \quad (5)$$

where D denotes the Hausdorff-Besicovitch dimension (sometimes called the "Hausdorff dimension" or "fractal dimension"). (5) is equivalent to

$$D = \frac{\ln N}{\ln (1/r)} \quad (6)$$

"ln" means "logarithmus naturalis". $\ln(x)$ is the power to which the constant e ($= 2.718281\dots$) must be raised to equal x : $e^{\ln x} = x$. Generally, you may take a logarithm to an arbitrary base, as R.F. Voss does in [3], pp. 28-30 and 61-63.

It is easy to prove that $D \geq 0$. For D to be negative, either the numerator or denominator of (6) would have to be negative. But this never happens: $1/r$ equals b , the number of segments into which the initial gauge was divided, and $b > 1$, so $\ln(1/r) = \ln b > \ln 1 = 0$. $N > 0$ (otherwise we had no gauge!) and is an integer (the number of edges of our starting gauge), therefore $N \geq 1$, so $\ln N \geq \ln 1 = 0$. Since $\ln N \geq 0$ and $\ln(1/r) > 0$, we must have $D = \ln(N) / \ln(1/r) \geq 0$.

All right: we now have a beautiful, fractal concept of dimension. It is time to see how we can profit from it! Measuring simple objects does no harm: for the cube with edges of $1/3$ unit length, $r = 1/3$, $N = 3^3 = 27$, thus $D = \ln 27 / \ln 3 = 3$, as Euclid said!

A circle of unit radius can be approximated by regular polygons, as we all know from school, where our teacher tried to introduce π ($= 3.14159\dots$). So we start with our line unit —

the ruler of length one unit — and fit it into the circle six times. Thus $N = 6$ in the first step. Our polygon approximating the circle is a hexagon with each side of length one unit: $r = 1$.

In the next step we use a dodecagon (12 vertices). To achieve this we have to choose $r = \sqrt{2-\sqrt{3}}$ ($= 0.51764\dots$) and increase N to 12. This process must be continued indefinitely, but we know what we obtain: 2π as the length of the perimeter. The length of the polygons is easy to calculate: Nr — the number of segments times the length of each one. Thus $r = 2\pi/N$. Now (6) yields:

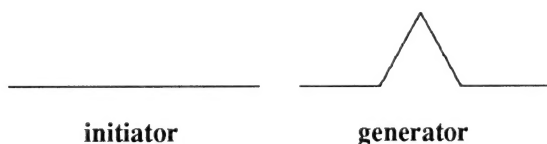
$$fN = \frac{\ln N}{\ln(1/r)} = \frac{\ln N}{\ln(N/2\pi)} = \frac{\ln N}{\ln N - \ln 2\pi} \quad (7)$$

Since approximating the circle is an infinite process, we need to allow N to increase to infinity. But then what becomes of our fraction? It approaches $\ln N / \ln N$, and so takes the value 1! If you don't believe me or don't understand why, I must disappoint you: this is not the right place for the fundamentals of analysis. Sorry! We know what we have, the Hausdorff-Besicovitch dimension of the circle is one, because

$$D = \lim_{N \rightarrow \infty} fN = 1 \quad (8)$$

Seems as if something went wrong when Euclid asserted the circle to be two-dimensional — or did he?

We see that with harmless objects, D is identical to D_t . We now have to refer to the terms “initiator” and “generator” of a fractal, in order to get an easier way to calculate D than using this trickery with approximation and limits. The starting object, which we intend to deform, we call the **initiator**. In the case of the Arc this would be a straight line segment. The object (with appropriate size reduction) which replaces the segments of the initiator we call the **generator**.



If these two factors, the initiator and the generator, contain enough information to construct the whole fractal, why should they not suffice to calculate D ? In the case of Koch's Arc, at every step we have to do four times as much work (replacing segments with a reduced copy of the generator) as in the previous step. To speak in our new terms, $N = 4^n$, for the Arc of the n -th degree. What happens to the size of the generator? In the first step (degree zero) the generator had its original size. To get the Arc of the first degree, we reduce the generator to one third of its original size, then (for the second degree) to one ninth, and so on. Thus for the n -th degree, the generator is $(1/3)^n$ of its original size. But this is just r , the length of the new unit we were just talking about. Let's see what this produces, when we put these values into (6):

$$D = \frac{\ln N}{\ln(1/r)} = \frac{\ln 4^n}{\ln 3^n} = \frac{n \ln 4}{n \ln 3} = \frac{\ln 4}{\ln 3} \quad (9)$$

$$= 1.26185907\dots, \text{ where } n > 0$$

Finally, a dimension which is a fraction! This is the [Hausdorff-Besicovitch] dimension of Koch's Arc, because it does not depend on n any longer. Don't confuse the Arc of the n -th degree with the Arc itself!

We learned that the result of the measurement depends on the quality of the gauge we have. For small n there would be a better gauge to use, but if we observe the process which takes place when n increases to infinity, our gauge becomes better and better. Since $(1/3)^n \rightarrow 0$ as $n \rightarrow \infty$, it cannot be improved, it is finer than any particular number greater than zero could possibly measure.

The rule that we derived is as follows:

“To calculate the Hausdorff-Besicovitch dimension of a fractal constructed with a generator and initiator, determine the ration of the measures of generator and initiator, take the logarithm of the numerator and of the denominator, then divide the first by the second.”

As this seems to work so well, we simply calculate D for another fractal:

For the Cantor Dust (Cantor set), the initiator is the unit

initiator

generator

interval (length one), and the generator is the first and last third of that interval, with total length $2/3$. Thus the ratio is $2:3$, and we get $D = \ln 2 / \ln 3 = 0.6309\dots$

If we compare the different dimensions of Cantor's dust, we see that $D_t = 0$ (it is totally disconnected), $D = 0.63\dots$, which $E = 1$. This is what B.B. Mandelbrot calls “dimensionally discordant” (see [1] p. 14). It is an example of the fact that

$$0 \leq D_t \leq D \leq E \quad (10)$$

Now we are prepared to understand Mandelbrot's definition of “fractal”:

A fractal is by definition a set for which the Hausdorff Besicovitch dimension strictly exceeds the topological dimension. ([1] p. 15)

Don't confuse *fractal* and *fractional* here! The Hausdorff-Besicovitch dimension of a fractal is not necessarily a fraction. Sets do exist where $D = 2$, but they are considered fractal because $D_t = 1$ (e.g. Sierpinski's Tetrahedron).

I must admit that the Hausdorff-Besicovitch dimension really is difficult to understand — much more difficult than the topological dimension. It is hard to determine the dimension of fractal sets, where information about the generator and the initiator is not as obvious as in our examples. Those fractals we encountered in this section are fractal *a priori* (to use Kant's favorite expression), meaning that they are fractal by the way they were created. But what if we discover something that seems to be fractal because at any magnification

we see more complex structures? I'll bet you anticipate that I'm thinking of the Mandelbrot set. Unfortunately I have no idea how to calculate its D; maybe someone else does... I really would not be too surprised to find that it lies strictly between one and two. That is what we all assume and expect.

This essay is an excerpt from
SUNRISE — The Ultimate Mandelbrot Set Editor, User's Manual and Tutorial, (c) 1988 Daniel Beisenherz • Dachauer Str. 258 • D-8000 München 50 • West Germany

References

- [1] B.B. Mandelbrot, "**The Fractal Geometry of Nature**", W.H. Freeman, NY (1983).
- [2] W.A. McWhorter & Jane Morrill Tazelaar, "**Creating Fractals**", *BYTE* (August 1987) 123-131.
- [3] Rainer Urian, "**Wir bauen uns ein Monster**", c't (May 1988) 146-158. (Turbo Pascal listing included).
- [4] H.-O. Peitgen, D. Saupe (eds.), "**The Science of Fractal Images**", Springer-Verlag, NY (1988).
- [5] M.G. Barrat & M.E. Mahowald (eds.), "**Geometric Applications of Homotopy Theory I**", Springer-Verlag, NY (1988).

PHOTOGRAPHING THE COMPUTER SCREEN

—Leonard Herzmark

Making slides or prints of your display is not a difficult feat, once you have settled on the color combinations you like. There are, however, a few details which must be attended to: the proper selection of film (be it slide or print), use of an appropriate filter for color, and, of course, you will need a camera.

A medium speed film like Eastman Plus-X Pan (ISO 125) is recommended for black and white, but you'll want to use color for maximum effect. I have found that the VRG 100

film is very effective for prints, and while I haven't used any, Eastman recommends Ektachrome for slides. I have used, with success, another film that is available from a limited number of suppliers: it is 5247, an Eastman movie film re-spooled for 35mm cameras. For slides, the processing service from the supplier in Los Angeles is very slow, so I will try the Ektachrome for my next slides.

The table at the bottom of this page shows Exposure Data for various films and camera types, as well as for color or monochrome monitors.

The only source for the filters that I was able to find was Spiratone, Inc. of Flushing, NY (800/221-9695). They were able to supply the appropriate filters and holders, and were very helpful. As long as I could give them the proper lens size for my camera — and of course my Master Card number, that was all that was necessary.

Filters and parts required for adapting to your camera:

SPIRA-

STONE #	DESCRIPTION	COST
386367	CC 20B 75mm sq gelatin filter	\$9.50 ea
386081	CC 40R 75mm sq gelatin filter	9.50
395005	SFS Filter holder cap	2.95
-	SFS Adapter ring (depends upon your camera). Price range: \$2.75-\$14.95.	
394122	Gelatin holder (2 required)	8.50 pair

You will need a tripod or other mount to hold the camera while the exposure is being made. For best results, the exposures should be done in a darkened room. The only source of light should be the monitor. Utilize the CC20B filter for shots that are predominantly blue, and the CC40R for shots that are predominantly red. If the color scheme is rather mixed, you need to experiment, but I have found that either filter works about as well.

My camera/lens configuration consists of a Minolta Maxxum 7000 with a 35-105 zoom (macro) lens. This enables me to get about 18 inches from the monitor to fill the frame.

Unfortunately, the aspect ratio of the screen does not conform to the format of a 35mm camera, so frame the screen, getting the top and bottom of the screen to fill the frame, and let the sides fall where they may. Of course, shoot to the left or right edge of the screen if you need to for your particular display.

FILM	COLOR MONITOR		B&W MONITOR	
	FOCAL-PLANE SHUTTER	LEAF SHUTTER	FOCAL-PLANE SHUTTER	LEAF SHUTTER
VRG-100 (Print) EKTACHROME 100 (Slide)	1/8 Sec @ f4.5	1/30 Sec @ f2.8	1/8 Sec @ f2.8	1/8 Sec @ f2.8
PLUS-X B&W (Print)	1/8 Sec @ f8	1/30 Sec @ f4	1/8 Sec @ f8	1/8 Sec @ f4

Focus sharply, use a cable release or the self-timer to avoid camera shake, be sure the lights are out, and shoot.

Cameras and lenses vary widely from owner to owner, so these instructions are very generalized. It will be necessary to experiment somewhat — what computer buffs don't?

Leonard E. Herzmark, P.E.

7417 Camello Del Norte #156

Scottsdale, AZ 85258

AUTHORS

—Ken Philip

Ph.D. Yale University 1963 (Radio Astronomy). Came to Fairbanks in late 1965 to work in solar radio astronomy at the Geophysical Institute, University of Alaska, Fairbanks. At the same time began studying Alaskan butterflies and moths, which developed into a part-time association with the Institute of Arctic Biology (and also became a Research Associate at the National Museum of Natural History, Smithsonian Institution, in 1977).

In 1975 my interests had shifted to the point that I switched from the Geophysical Institute to the Institute of Arctic Biology, and began devoting my full time to the Alaska Lepidoptera Survey. The ALS collection (now over 86,000 specimens) forced me to develop an interest in microcomputers as aids to collection management, and in 1981 I acquired an Apple II+. Things sort of evolved from there, culminating with the acquisition of a Mac II last summer.

Since 1970 I have been going on one major collecting trip each summer to remote northern areas. The high points of my field career have been my three summers of collecting in the Magadanskaya Oblast', northeastern Siberia, USSR (1978, 1980, 1983) under the auspices of the scientific exchange program between the National Academy of Science and the Soviet Academy of Science — but I have also spent several summers on the Seward Peninsula, North Slope and Brooks Range, and in the northern Yukon Territory, as well as one summer on Victoria Island in the Canadian Arctic Archipelago.

Current affiliations are: Research Associate, Institute of Arctic Biology, University of Alaska, Fairbanks.

Research Associate, Department of Entomology, National Museum of Natural History, Smithsonian Institution, Washington, D.C.

QUESTIONS

Does anyone have information on Harmonic Analysis using fractals?

C.J. Vaccare • Rt 1 Box 476-B • Crewe, VA 23930

I'd like to know if it's possible to get a gridded wall chart of $\mathcal{M}$.

Joel Kahn • 534 E. Grand • Springfield, MO 65807

PRODUCTS

If you place an order as a result of seeing the following notice, please mention Amygdala with your order.

PRINTING $\mathcal{M}$ POSTERS

I have written a program (Turbo Pascal 4.0) for printing the Mandelbrot Set on an IBM PC with a matrix printer (Epson MX82). There are sure a lot of such programs around, but I wanted one to print posters of the Mandelbrot Set in any size. As you can see from the included pictures, the quality on a 9 dot matrix printer is acceptable.

I'd like to share this program with interested people for a copy and shipping charge of DM 90.-- (Deutschmarks).

The program expects the desired range as input and calculates the corresponding Mandelbrot Set, which is saved on disk. Calculation may be interrupted at any time by the user and can be continued later on. This enables you to use your office PC during breaks and through the night, for you know that big pictures take a long time.

When a picture's calculation is done, it can be printed as often as desired — and in different selectable grey values (overstrike method) — on the printer. Pictures can be as big as you specify — you can print more than one page and tape the pages together. I have a Hercules graphics card (black and white) and provide output for this type of card on the screen.

Turbo Pascal 4.0 supports other graphics cards (CGA, MCGA, EGA, VGA, AT&T 6300, IBM 3270 PC) in a device independent manner, so there should be no problem in using them.

Thomas Sigl • Hartmannstr. 89 • D-8520 Erlangen
WEST GERMANY

RS comments: Thomas enclosed 3 samples of the output of his program. They are all remarkable. The largest is a 31"x31" view of a double spiral with remarkable detail of substructure. I'd estimate the pixelation at about 2400 x 2400.

ART MATRIX; PO Box 880; Ithaca, NY 14851 USA. (607) 277-0959. "Nothing But Zooms" video, Prints, FORTRAN program listings, postcard sets, slides. Send for FREE information pack with sample postcard. Custom programming and photography by request. Make a bid.

CIRCULATION

As of November 3, 1988 Amygdala has 509 paid-up subscribers (for the new series), 215 of whom have the supplemental color slide subscription.